Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks

Rishabh Madan*, Deepank Agrawal*, Shreyas Kowshik*, Harsh Maheshwari*, Siddhant Agarwal*
and Debashish Chakravarty
Autonomous Ground Vehicle Team, Indian Institute of Technology, Kharagpur, India

Keywords: Convolutional Neural Network, Feature Descriptors, Traffic Sign Classification, Histogram of Oriented Gradient, Speeded Up Robust Features.

Abstract: Traffic signs play an important role in safety of drivers and regulation of traffic. Traffic sign classification is thus an important problem to solve for the advent of autonomous vehicles. There have been several works that focus on traffic sign classification using various machine learning techniques. While works involving the use of convolutional neural networks with RGB images have shown remarkable results, they require a large amount of training time, and some of these models occupy a huge chunk of memory. Earlier works like HOG-SVM make use of local feature descriptors for classification problem but at the expense of reduced performance. This paper explores the use of hybrid features by combining HOG features and SURF with CNN classifier for traffic sign classification. We propose a unique branching based CNN classifier which achieves an accuracy of 98.48% on GTSRB test set using just 1.5M trainable parameters.

1 INTRODUCTION

Traffic sign classification holds an essential place in visually guided driving assistance and autonomous driving systems and several other traffic-related utilities. Traffic signs are utilised as a method of warning and guiding drivers, helping to regulate the flow of traffic among vehicles, pedestrians, and others who travel the streets, highways and other roadways. The development of traffic sign classification is dedicated to reducing the number of fatalities and the severity of road accidents and is an important and active research area. In general, traffic signs have unique and distinctive features like simple shape priors in the form of circles and triangles combined with uniform colouring patterns, which makes them easily recognizable and thus also a restricted classification task. Regardless, classifying these signs without any human supervision is still a challenging task considering the different kinds of problems like occlusions, disoriented poles, lighting changes and poor quality signs that are encountered during real-world execution.

For many years, local features descriptors have dominated all domains of computer vision. The considerable progress that has been visible in classification and detection is mostly due to them. Histogram of Gradients (HOG) after its introduction, outperformed all the previous methods for human detection (Dalal and Triggs, 2005) by a considerable margin. It has since then been used to solve a variety of classification and detection problems. (Schmitt and McCoy, 2011) used Speeded Up Robust Features (SURF) (Bay et al., 2006) feature descriptors along with Support Vector Machine (SVM) classifier for classification on a subset of the CalTech-101 database (Fei-Fei et al., 2004).

In recent years, convolutional neural networks (CNN) have become pervasive in classification tasks after AlexNet (Krizhevsky et al., 2012) which popularized deep convolutional neural networks for classification. The general trend has been to make deeper and more complicated networks to extract high-level features in order to achieve higher accuracy (Szegedy et al., 2014),(He et al., 2015). Many approaches involving deep learning have achieved exemplary performance in traditional road environments. However, an increase in depth of network leads to exponential increase in the number of parameters. A deep convolutional neural network involving Spatial Transformer networks (García et al., 2018) achieved highest accuracy on the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2012) dataset

*These authors have contributed equally.

Madan, R., Agrawal, D., Kowshik, S., Maheshwari, H., Agarwal, S. and Chakravarty, D.
Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks.
DOI: 10.5220/0007392506130620
In *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pages 613-620
ISBN: 978-989-758-351-3

but used 14M model parameters. These advances to improve accuracy are making networks inefficient in terms of memory use and inference speed. In the real world, traffic sign classification might be carried out on a memory constrained platform but would still require highly accurate prediction.

In this work, we propose the use of hybrid feature descriptors along with a CNN based classifier for traffic sign classification thereby providing a computationally efficient method with a lesser number of model parameters and improved training time. The proposed hybrid feature descriptors comprise of HOG feature descriptors and Bag of Words (BoW) variant of SURF feature descriptors. HOG descriptors tend to capture the global features of the object and are lighting invariant, thus beneficial in the case of traffic signs, due to the diverse shapes and high-level geometric priors for different signs. But HOG descriptors highly vary with change in object orientation, which could become a problem when these signs are viewed from multiple viewpoints. On the other hand, SURF descriptors are rotation invariant, but when using a BoW representation, the spatial and geometric relationship information between descriptors is lost. This motivates the use of a combination of HOG descriptors with SURF BoW descriptors. The presented approach leverages the use of these hand-crafted descriptors by using a CNN to learn a better representation for finer feature extraction and observe performance comparable to recent deep CNN architectures using RGB images for classification. The use of this combination of descriptors to classify traffic signs using a basic CNN classifier has been demonstrated. The purpose of using CNN, in comparison to SVM or other learning-based classifiers in this particular case is to learn a better feature representation of this combination of two different descriptors and thus make use of relatively high-dimensional features for classification. We extend the experiments by using a branched CNN architecture to reduce the number of trainable model parameters by a large factor and also observe incremental improvements in terms of classification accuracy.

Contributions. The main contributions of this paper comprise the use of a hybrid combination of HOG and SURF feature descriptors as an input to the CNN classifier for accurate classification of traffic signs. We perform an extensive comparative study regarding the combined use of these two very different feature descriptors, and how they support each other to improve the prediction results with minimal training parameters by a substantial margin.

2 RELATED WORK

HOG descriptors with Support Vector Machine (SVM) classifier have been used for classification purposes earlier (Blauth et al.,). Even though these features show better performance in characterizing object shape and appearance, there is a drawback of this approach that it is restricted to binary classification as the SVM determines the optimal hyperplane, separating two classes in the dataset. The use of SVM in a multinomial classification problem thus becomes a case of one-versus-all, in which the positive class is the class with the highest score whereas the rest represent the negative class. In other words, we need to train N-SVMs for an N-class classifier, whereas an N-class classifier neural network can be trained in one go. Also, the neural network can generalize in a better manner as it acts like one whole system whereas SVMs are isolated systems.

SURF descriptors with CNN have been previously used by (Elmoogy et al., 2018) for solving the problem of indoor localization. CNNs are capable of extracting high-level features but require high dimensional optimization procedure due to which the training time is significantly long. On the other hand, image descriptors obtain features from the image through deterministic means which have much higher speed than CNN. The disadvantage of the descriptor is that the output feature size is generally large compared to CNN. In this approach, they combined both the feature descriptor and the CNN, by first using an image descriptor to extract features from the image and then using the CNN to reduce the dimension of the feature. SURF features alone are not able to represent the geometric property of the image. This complication is dealt with using combined HOG features with SURF.

(Abedin et al., 2016) have taken a similar approach of using hybrid features for traffic sign classification. They have used SURF and HOG feature descriptors with Artificial Neural Network (ANN) for classification but have not shown their methodology regarding how they are combining both descriptors. They do not provide any valid reasons for the use of this combination and do not explain why using individual HOG, or SURF descriptors would not work. Also, they have tested their approach for classifying just 4 different traffic signs on a very small dataset which makes their approach unreliable in case of large and complex datasets. In contrast, we establish proper reasoning and show detailed experiments on 43 different traffic signs, concerning the use of the hybrid feature. Moreover, a CNN is used instead of a Multilayer Perceptron (MLP). CNN works particularly well on data having a spatial relationship and is thus ideal for the task of traffic sign classification.

3 METHOD

In this paper, a method involving the use of multiple feature descriptors with CNN for image classification is introduced. This method is applied to the problem of traffic sign classification. The procedure begins by first generating the HOG descriptor for the entire dataset followed by extraction of SURF that have high saliency and lastly feeding features to a convolutional neural network for prediction.

3.1 Feature Extraction with HOG

HOG is a dense feature extraction method. It detects complex shapes of structures by the distribution of intensity gradients or edge directions. HOG generates pixel-wise histograms of gradient directions and concatenates them to get the descriptor.

HOG features are based on magnitude distributions and gradient angles. Due to this, they have a natural invariance to changes in lighting conditions and colour variation. These make them robust in visual data. It involves first computing the gradients of all pixels. For an image I, gradient estimation filters as $H_x = [-1,0,1]$ *and* $H_y = [-1,0,1]^T$. Let G_x *and* G_y be the gradient matrices generated by

$$G_x = I * H_x \text{ and } G_y = I * H_y \tag{1}$$

where * is the convolution. The gradient value at each pixel can be calculated as

$$g(i,j) = \sqrt{(G_x(i,j)^2 + G_y(i,j)^2)} \tag{2}$$

and the dominant gradient direction at each pixel can be estimated by

$$\theta(i,j) = tan^{-1}\left(\frac{G_y(i,j)}{G_x(i,j)}\right) \tag{3}$$

This is followed by creating cell histograms. Each point in a cell casts a weighted vote for the histogram channel. These votes are based on the gradient values computed earlier. This orientation based channel is distributed over 0 to 180°. The cells are grouped into spatially connected larger blocks and normalized locally providing an invariance in changes in illumination and contrast. These normalized cell histograms are concatenated to form the HOG feature vector.

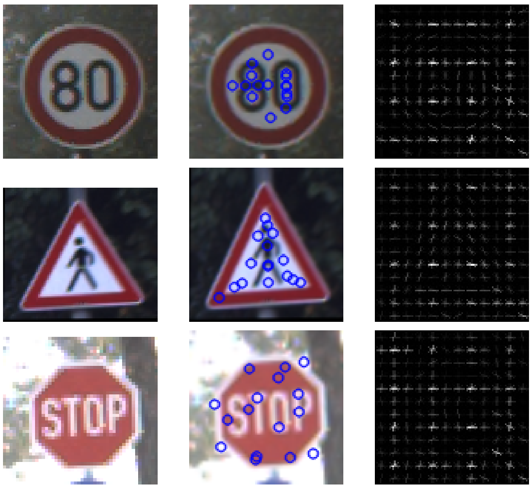

Figure 1: The above figure shows the image in column (a) and their corresponding SURF and HOG features in column (b) and column (c).

3.2 Feature Extraction with SURF

SURF is an interest point detector and descriptor. It is scale and rotation invariant and thus is more reliable for practical purposes since camera feed mostly provides tilted and scaled traffic signs.

To calculate the orientation of a point, SURF uses wavelet responses in horizontal and vertical directions for a neighbourhood of size 6. The sum of all responses within a sliding orientation window of angle 60°is calculated. This gives us the dominant orientation. The detector is based on the determinant of the Hessian matrix.

Given a point $x = (x,y)$ in an image I, the Hessian matrix $\mathcal{H}(x,\sigma)$ in x at scale σ is defined as follows

$$\mathcal{H}(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{4}$$

where $L_{xx}(x,\sigma)$ is the convolution of the 2^{nd} order derivative of Gaussian with image I in point x and same for $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$.

SURF uses box filters as an approximation to Laplacian of Gaussian(LoG) for computational advantages. The descriptor, on the other hand, describes a distribution of Haar-wavelet responses within the interest point neighbourhood. SURF is sensitive to lighting conditions and image-distortions thus producing variable distributions of feature vectors across the dataset. To ensure a fixed dimensionality of the SURF feature vectors, descriptors were clusteresd using K-Means algorithm.

Traffic signs comprise of simple geometric patterns. Using only SURF BoW descriptors for classification will not work as SURF BoW does not store the information related to geometric patterns. HOG features, on the other hand, identify these simple geometric patterns very well. But there is one potential problem with using only HOG features. HOG features are highly invariant to changes in lighting conditions but get hugely affected by orientation changes of the sign and do not detect local features. In real world, the images of traffic sign come from different viewpoints and may also be slightly distorted. So, HOG alone cannot accurately classify traffic signs. For this purpose, SURF BoW descriptors are used, which are rotation and orientation invariant and capture the local features. HOG detects the basic geometry of the sign while SURF BoW descriptors complement HOG by making it more robust to changes in orientation and capturing local features.

Algorithm 1: High saliency SURF feature extraction.

```
procedure SURF_GENERATE(image, num_clusters)
    $\mathcal{H} \leftarrow 0$
    while True do
        SURF($\mathcal{H}$)
        $m = f(image)$
        if $num\_feature_x < \varepsilon_{min}$ then
            skip the image
        if $num\_feature_x > \varepsilon_{max}$ then
            $\mathcal{H} \leftarrow \mathcal{H} + \alpha$
        else
            break
    SURF($\mathcal{H}$)
    $m = f(image)$
    KMEANS($x$, num_clusters)
                                        ▷ Cluster m along x axis
    return cluster_centers
```

Where $\mathcal{H}$ represents Hessian Threshold for SURF feature. α is step size with which $\mathcal{H}$ is increased. ε_{min} and ε_{max} represent the minimum and maximum threshold values of number of feature along x-axis, between which loops break and clustering is done.

3.3 Branched Pipeline

Initially, a generic CNN architecture is used as a classifier. This classifier was directly fed HOG features appended with SURF. While it achieves good results on the GTSRB test set, it has a large number of trainable parameters and takes a lot of time for training. To solve this problem, a unique branched CNN architecture is proposed. This model uses HOG features and SURF as input to two different branches that consist of convolutional blocks similar to the generic CNN. The embeddings obtained from these two branches are concatenated and further passed through fully connected layers. This reduces the model parameters, thus reducing computational cost. Additionally, improvement is also observed in the accuracy on the test set. This can be attributed to the fact that different convolutional filters are used for the two different kinds of feature descriptors.
For a detailed analysis of the results of the descriptors, refer to the Analysis Section 5.

4 EXPERIMENTS

4.1 Dataset

German Traffic Sign Recognition Benchmark (GTSRB) dataset (Stallkamp et al., 2011) has been used to perform all the experiments. It consists of 39,029 training images spread across 43 classes. The distribution of classes is highly skewed with around 200 in one up to 2000 in another class. The dataset has been created by extracting image frames from 1-second video sequences. A single sequence of 30 images usually contains images of the same traffic sign with increased size. Thus it is important to employ a proper strategy to create a meaningful training dataset. The technique used by (Sermanet and LeCun, 2011) is used to tackle the above issue.

To bring in uniformity, all classes are augmented by applying a random brightness, random rotation and random distortion to the image and those images are flipped which are invariant to horizontal, vertical and 180°flips. This is done for all the classes. This creates a dataset of over 130k images where data distribution across different classes is still skewed. To tackle this problem, further augmentation is done in selected classes with less than 1000 images by applying a random translation to them. The number of augmented images of other classes was reduced to around 1200. This creates a comparatively uniform distribution (Figure 2) after applying HOG and SURF to the images with 53,238 images in the dataset.

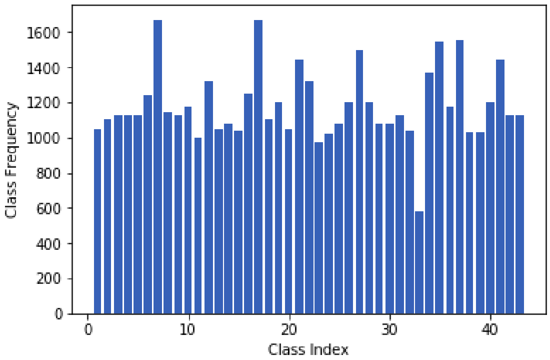

Figure 2: No. of training images of each class.

The HOG and SURF vectors are computed on each image (refer to Appendix for function parameter values). Each SURF vector in its 64-dimensional space is clustered into 8 clusters using KMeans. This gives us a 1x1764 sized HOG-vector. and 1x512 sized SURF-vector.

4.2 Network Architectures

We broadly describe the architectures used for performing all the experiments. For complete implementation details, refer to the Appendix 6.

4.2.1 Basic CNN Architecture

Figure 6 [in Appendix 6] shows the preliminary CNN architecture that is used as a classifier along with extracted features as the input. The network comprises of two convolutional blocks, where each block consists of a batch normalization layer, ReLU activation and max pooling layer. The embedding after passing the input features through them is flattened and fed to fully connected layers, and finally predicted probabilities for all 43 classes are returned in the final layer.

4.2.2 Branched CNN Architecture

The basic CNN model suffers from the problem of a large number of model parameters. To improve its performance on computationally limited resources and reduce the model parameters a branched CNN (Figure 7) architecture is used, where the input HOG and SURF are separately fed into two different branches, each consisting of two convolutional blocks similar to the ones used in the basic architecture. The respective embedding received from these two branches are flattened, concatenated and passed through fully connected layers to output the individual probabilities for each class. The HOG conv-branch is fed input HOG features of shape 7x252, and the SURF-branch is fed input of 8x64. This method of branching allows us to apply convolutions on the two different type of feature maps separately. As can be seen in Table 1, the number of model parameters is reduced by 6x when using the branched CNN model.

Table 1: Comparison of number of model parameters.

Model Architecture	No. of Parameters
Basic CNN	8,543,467
Branched CNN	1,583,947
3-Conv-No-STN	7,303,883
3-Conv-3-STN	14,629,801

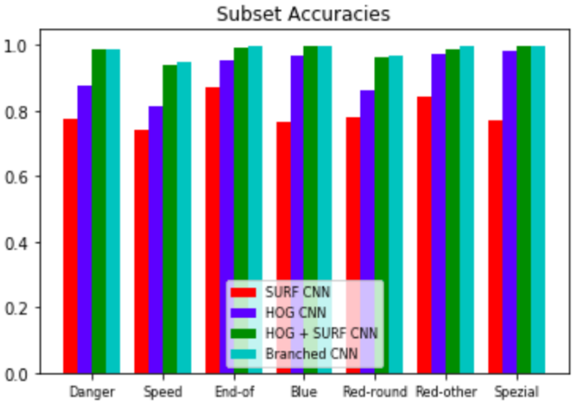

Figure 3: Comparision of subset accuracies.

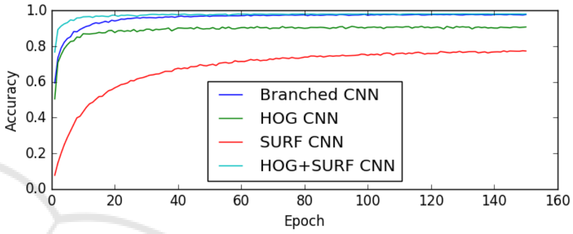

Figure 4: Comparison of test accuracy for the experiments.

5 RESULTS

To validate the use of multiple feature descriptors with a CNN based architecture, the Basic CNN architecture is used. Upon experimenting with a branched CNN architecture, better accuracy is observed. over the basic architecture. Figure 4 shows the test accuracies of different models during training. Table 3 shows the accuracy obtained on the test dataset. Figure 3 compares the subset accuracies for different experimental models. The different subsets consist of “Danger” signs, “Speed” signs, “End-of” signs, “Blue” signs, “Red round” signs, “Red other” signs, “Spezial” signs as defined in the GTSRB dataset (Stallkamp et al., 2011). The branched CNN architecture with multiple features outperforms the HOG-SVM based approach. The model performance is compared with (García et al., 2018) which shows state of the art results on GTSRB dataset. Their base model is defined as 3-Conv-No-STN, i.e. CNN without spatial transformer networks (STN) and their best model as 3-Conv-3-STN, i.e. CNN with three STNs. It is visible from Table 1 and Table 3 that even though state of the art method achieves a slightly better accuracy, due to large number of trainable parameters it is heavily computation intensive during train and test time when compared to the proposed method.

Table 2: Analysis of False Positive count (FP) for some high FP struck classes. Third Column is the class that is predicted with maximum FP count. The class indices correspond to the GTSRB dataset.

Class Index	Total FP		Class Most Confused With	Corresponding FP	
	HOG	HOG + SURF	HOG	HOG	HOG + SURF
Pedestrians Possible	40	5	Danger Point	18	0
Road narrows on the right	72	1	Danger Point	45	0
End of (truck)	45	2	End of (car)	44	1

Table 3: Test set accuracy for different classifiers and inputs.

Classifier	Input Data	Accuracy(%)
Basic CNN	HOG	91.09
Basic CNN	SURF	77.41
SVM	HOG	96.93
Basic CNN	HOG + SURF	98.07
Branched CNN	HOG + SURF	98.48
3-Conv-No-STN	RGB Image	98.81
3-Conv-3-STN	RGB Image	99.49

Using HOG features with the branched model yields an accuracy close to 90%. After checking the model outputs on classes having low test accuracy, it is observed that the model confuses it with traffic signs having a similar shape but does not distinguish well enough between the drawing placed inside the shape. This validates the theory that HOG features lack information about local features. Experiments are also performed by just feeding the SURF to the basic model and a poor accuracy of 77.41% is observed on test set. Using both the HOG and SURF with the branched CNN model outperforms all the baselines using feature descriptor as input and achieves comparable accuracy with respect to state of the art method.

Analysis. Refer to Table 2 for the following analysis. It corresponds to classes represented by images in Figure 5. With reference to this figure, it is evident that HOG features tend to capture the overall shape of an object very well, leaving apart the intricate details embedded in the shapes. This property of HOG features causes it to confuse the images in the left column with the images in the right column as evident in Table 2. SURF, on the other hand, describes localized key-points embedded in the shapes. Upon using HOG and SURF together, SURF entirely learned the unique features of the ”danger point” sign (Figure 5) and brought down the total FP from 40 to 5. Similar improvements were observed in case of ”Road Narrows on the right” (row 2, Figure 5). SURF learned to distinguish between the two different “End of” signs shown in Figure 5, by bringing down the FP count from 44 to just 1. As can be seen, the two classes are very similar in appearance and only a few unique key points can differentiate between them, which SURF captured successfully.

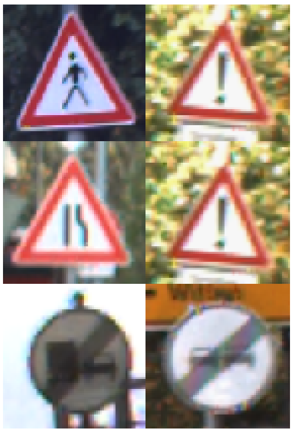

Figure 5: Left - Classes with maximum false positives with only HOG features (Ground Truth Images). Right - Predicted class with only HOG features.

6 CONCLUSION

We proposed the method of concurrently using the HOG and SURF of an image with a CNN based architecture for the classification of traffic signs in the GTSRB traffic sign dataset. The proposed pipeline using the basic architecture achieved an accuracy of 98.075%. The performance of the approach is further enhanced by using a branched CNN architecture achieving an accuracy of 98.48% on the test set. The advantages of using two different feature descriptors, HOG and SURF together, are examined over previous works that either use a RGB image or the above-mentioned features individually. The experiments show that using these pre-computed hybrid features along with CNN achieves slightly lower performance to the state of the art method but at the gain of much lesser number of parameters, hence leading to reduced computational resource usage by manifolds. In future work, we intend to focus on using the proposed technique of traffic sign classification with the existing region proposal networks (RPN) to perform efficient real-time detection of traffic signs. We also intend to fine tune this method for classifying Indian traffic signs.

REFERENCES

Abedin, M. Z., Dhar, P., and Deb, K. (2016). Traffic sign recognition using hybrid features descriptor and artificial neural network classifier. In *2016 19th International Conference on Computer and Information Technology (ICCIT)*, pages 457–462.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg. Springer Berlin Heidelberg.

Blauth, M., Kraft, E., Hirschenberger, F., and Böhm, M. Large-scale traffic sign recognition based on local features and color segmentation.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Elmoogy, A. M., Dong, X., and Lu, T. (2018). Cnn : a descriptor enhanced convolutional neural network.

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178.

García, Á. A., Álvarez, J. A., and Soria-Morillo, L. M. (2018). Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural networks : the official journal of the International Neural Network Society*, 99:158–165.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Schmitt, D. and McCoy, N. (2011). Object classification and localization using surf descriptors.

Sermanet, P. and LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2809–2813. IEEE.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). The german traffic sign recognition benchmark: a multi-class classification competition. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1453–1460. IEEE.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.

APPENDIX

HOG features and SURF are computed using the OpenCV library. The parameters used during HOG computation are window size = (32,32), block size = (8,8), block stride = (4,4), cell size = (4,4), number of bins = 9. All experiments were performed using Tensorflow. Dropout was used in all convolutional layers with a probability of 0.6. Adam Optimizer, with an initial learning rate of 1e-4 and default parameters, was used.

Basic Architecture

The basic architecture was implemented entirely in Tensorflow. With reference to Figure 6, Conv_C_K_S refers to a convolution layer with 'C' output channels, 'KxK' kernel size and an 'SxS' strided convolution. All Pool layers are Max-Pool layers with a kernel of 2x2 and stride of 2x2. This reduces the input size by a factor of 2 at each stage. A ReLU activation is applied after each block. We implement Batch Normalization after each convolutional block with the default scale and shifting parameters in tensorflow. Batch Normalization is not applied to fully connected layers. The output of the network is the Softmax activation probabilities over the 43 classes of the GTSRB dataset. The loss function used is as follows:-

$$\mathcal{L}(y,y') = -\sum y' log(y) \tag{5}$$

where y are the labels for classification and y' are the predictions made (the logits). This represents the cross-entropy loss for multiple classes.

Branched CNN Architecture

The Branched Architecture was implemented in a similar fashion to the Basic Architecture with similar default parameters. Batch Normalization was applied only after each convolutional block and not on any fully connected layer.

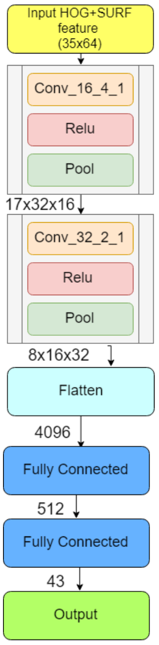

Figure 6: Basic CNN Architecture.

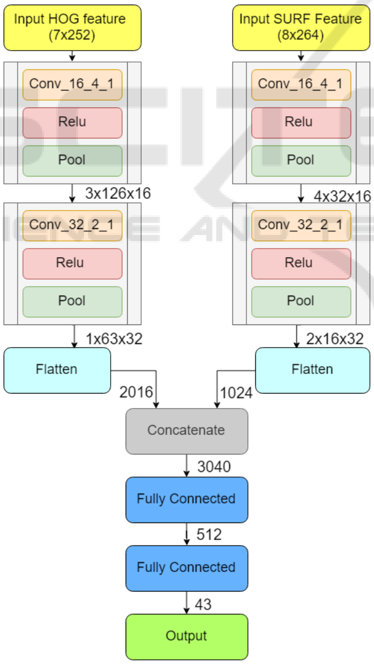

Figure 7: Branched CNN Architecture.